Übungen zur Vorlesung Einführung in die Informatik II

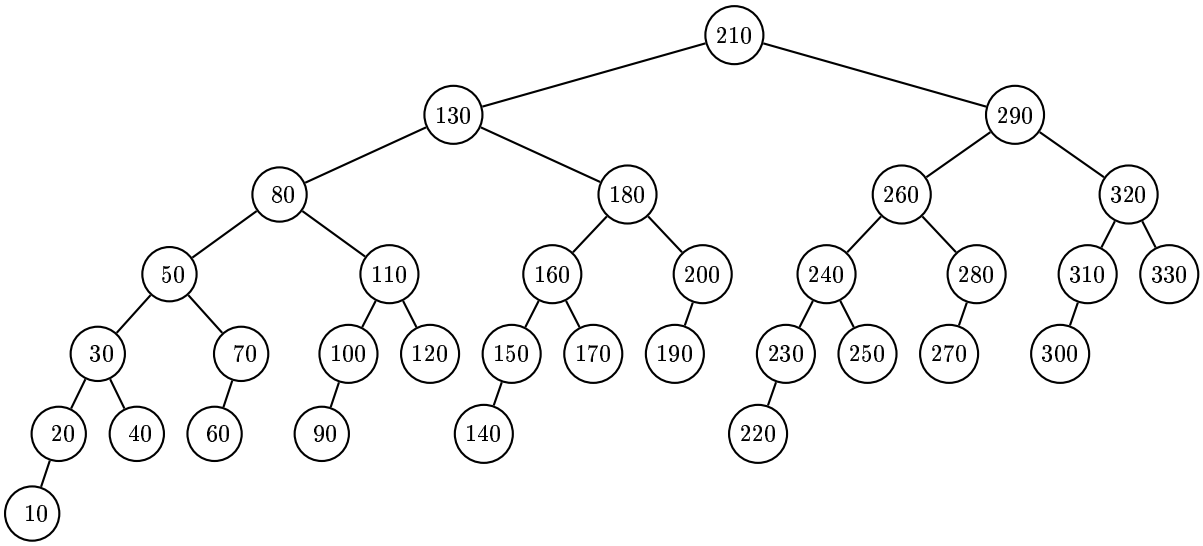
Ausgabe: 18. Juni 2002 Abgabe: 24. Juni 2002, 13.00 Uhr

Die Abgabe erfolgt ausschließlich über das System eClaus! Siehe
http://www.informatik.uni-stuttgart.de/ifi/fk/lehre/ss02/info_II_02.html

Nehmen Sie bitte an der Vorlesungsumfrage teil!
http://fachschaft.informatik.uni-stuttgart.de/VU/

Aufgabe 1: (AVL-Bäume) schriftlich

(3 Punkte) Gegeben sei der unten abgebildete AVL-Suchbaum. Löschen Sie den Knoten mit Inhalt 330. Geben Sie die zwischenzeitlich entstehenden AVL-Bäume an sowie den Ergebnisbaum nach der Löschung des Knotens mit Inhalt 330.



Aufgabe 2: (Eigenschaften von AVL-Bäumen) zum Votieren

(2 Punkte) Beweisen oder widerlegen Sie mit einem Gegenbeispiel folgende Behauptung:
Es existiert ein $\alpha > 0$, so dass jeder AVL-Baum ein α-gewichtsbalancierter Baum ist.

Aufgabe 3: (AVL-Baum-Variante) zum Votieren

AVL-Bäume haben die Eigenschaft, dass sich die Höhe der Unterbäume um jeweils maximal 1 unterscheidet.

a. (1 Punkt) Definieren Sie AVL2-Bäume. Dieses sind höhenbalancierte Suchbäume, deren Unterbäume sich in der Höhe um maximal 2 unterscheiden.

b. (2 Punkte) Überlegen Sie sich, welche Situationen beim Einfügen von Elementen entstehen können. Bei welchen Situationen muss rotiert werden (und wie), damit die Höhenbalanceeigenschaft (Höhen der Unterbäume unterscheiden sich um maximal 2) wieder hergestellt ist?

c. (2 Punkte) Überlegen Sie sich dies auch für das Löschen von Knoten.

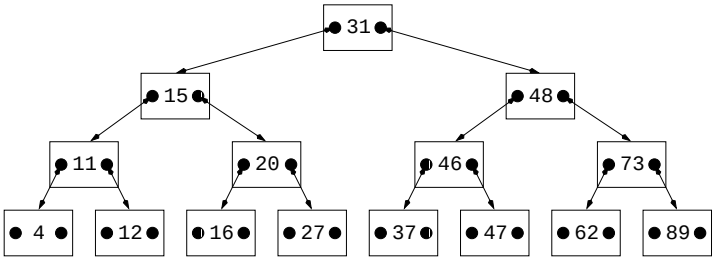
Aufgabe 4: (AVL2-Bäume) **schriftlich**

(4 Punkte) Welche Höhe kann ein AVL2-Baum mit n Knoten maximal haben? (Hinweis: Zeigen Sie, dass ein AVL2-Baum der Höhe $r+1$ mindestens $\left(\sqrt{2}\right)^r$ Knoten haben muss.)

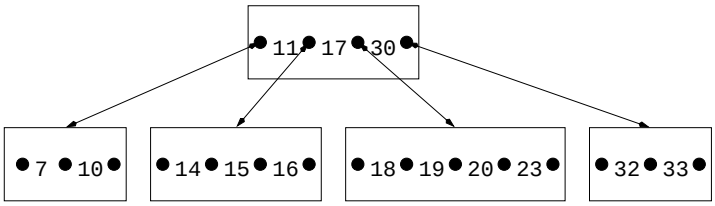
Aufgabe 5: (B-Bäume) **zum Votieren**

a. (1 Punkt) Fügen Sie in einen anfangs noch leeren B-Baum der Ordnung $m = 1$ nacheinander die folgenden Elemente in der angegebenen Reihenfolge ein: 10, 21, 14, 17, 88, 7, 42, 36, 75, 22, 54 und 1. Geben Sie dabei jeden durch das Einfügen eines Elementes entstandenen Baum an.

b. (1 Punkt) Führen Sie für folgenden B-Baum der Ordnung $m = 1$ die beiden Operationen "Löschen von 16" und "Löschen von 47" durch.



c. (1 Punkt) Führen Sie für den folgenden B-Baum der Ordnung $m = 2$ die Operationen "Einfügen von 24" und "Löschen von 10" aus.



Aufgabe 6: (Hashtabellen) **schriftlich**

(3 Punkte) In eine Hashtabelle mit 13 Einträgen (Indizes 0 bis 12) sollen die Wörter Borneo, Bielefeld, Finnland, Hamburg, Katmandu, Rabenstein, Schwerin, Spandau, Togo und Wuerzburg in der angegebenen Reihenfolge eingefügt werden. Kollisionen sollen mit dem Verfahren der linearen Sondierung aufgelöst werden. Die Schrittweite ist die Länge des einzufügenden Wortes. Die verwendete Hashfunktion ist:

```
h(w) = (2 * (Character'POS(w₁) - Character'POS('A'))
       + 3 * (Character'POS(w₃) - Character'POS('a'))) MOD 13
```

wobei w_i der i-te Buchstabe des Wortes ist. `Character'POS liefert` den ASCII-Code eines Zeichens zurück. Sollten Sie keine ASCII-Tabelle zur Hand haben, so lassen Sie sich mit diesem Programmfragment eine ausgeben:

```
for I in Character'RANGE loop
  Text_IO.Put_Line (I & " " & Integer'IMAGE (Character'POS(I)));
end loop;
```

Geben Sie die ausgefüllte Tabelle, sowie für jeden Eintrag alle die im Zuge der Kollisionsauflösung berechneten Indizes an.